TSGAssist: An Interactive Assistant Harnessing LLMs and RAG for Time Series Generation Recommendations and Benchmarking

Yihao Ang*
National University of Singapore
NUS Research Institute in Chongqing
yihao_ang@comp.nus.edu.sg

Yifan Bao*
National University of Singapore
yifan_bao@comp.nus.edu.sg

Qiang Huang†
National University of Singapore
huangq@comp.nus.edu.sg

Anthony K. H. Tung
National University of Singapore
atung@comp.nus.edu.sg

Zhiyong Huang
National University of Singapore
NUS Research Institute in Chongqing
huangzy@comp.nus.edu.sg

ABSTRACT

Time Series Generation (TSG) is essential in many industries for generating synthetic data that mirrors real-world characteristics. TSGBench has advanced the field by offering comprehensive evaluations and unique insights for selecting suitable TSG methods. However, translating these advancements to industry applications is hindered by a cognitive gap among professionals and the absence of a dynamic platform for method comparison and evaluation. To address these issues, we introduce TSGAssist, an interactive assistant that integrates the strengths of TSGBench and harnesses Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) for TSG recommendations and benchmarking. Our demonstration highlights its effectiveness in (1) enhancing TSG understanding, (2) providing industry-specific recommendations, and (3) offering a comprehensive benchmarking platform, illustrating its potential to ease industry professionals' navigation through the TSG landscape and encourage broader application across industries.

PVLDB Reference Format:
Yihao Ang, Yifan Bao, Qiang Huang, Anthony K. H. Tung, and Zhiyong Huang. TSGAssist: An Interactive Assistant Harnessing LLMs and RAG for Time Series Generation Recommendations and Benchmarking. PVLDB, 17(12): 4309 - 4312, 2024.
doi:10.14778/3685800.3685862

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/YihaoAng/TSGAssist/.

1 INTRODUCTION

Time Series Generation (TSG) has emerged as an indispensable technique across numerous industries. The essence of TSG lies in its ability to generate synthetic data that retains the intrinsic statistical properties and temporal dependencies of original data. This capability is crucial in domains such as finance for risk assessment, environmental science for climate modeling, and manufacturing for predictive maintenance, especially when real data is limited, sensitive, or unevenly distributed.

While TSG methods have proliferated, their practical use in industry is not straightforward, as each method comes with its own set of strengths and limitations. Identifying the optimal method for specific industrial applications is challenging. Moreover, assessing the quality of generated time series necessitates a suite of thorough measures that scrutinize various facets of data fidelity and temporal coherence. TSGBench [2] represents a significant leap in standardized TSG evaluation, offering a benchmarking framework with a broad range of comprehensive metrics. It further aids in choosing appropriate TSG methods and evaluation measures tailored to specific data characteristics and requirements.

Nevertheless, the translation of academic TSG advancements into industry practice faces notable challenges. First, there is a cognitive gap for industry experts: the sheer volume of TSG methods and metrics can be daunting, with understanding their nuanced differences and suitability necessitating extensive academic knowledge. Second, the lack of a dynamic platform hinders industry professionals from effectively exploring or comparing the performance of various TSG methods in real-world data contexts.

To tackle these challenges, we introduce TSGAssist, an interactive TSG Assistant that harnesses the core strengths of TSGBench, offering an intuitive, user-friendly interface for industry professionals. TSGAssist comprises two key components:

- **TSG Recommender**: Employs advanced Large Language Models (LLMs) [4] and Retrieval-Augmented Generation (RAG) [9] to provide personalized, context-aware recommendations for TSG methods and evaluation metrics. Its conversational interface bridges the knowledge gap, allowing users to interact naturally with the system and receive custom advice based on their specific industry needs and data characteristics.
- **TSG Benchmarking**: Enables users to apply recommended TSG methods to their data directly on the platform. It supports instant comparison and evaluation with a wide range of metrics, providing practical insights into the application of various methods on real datasets.

*Yihao Ang and Yifan Bao contributed equally to this research and were ordered alphabetically by their last names.
†Qiang Huang is the corresponding author.

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685862

Our demonstration showcases TSGAssist's effectiveness and ease of use through three scenarios: (1) enhancing real-world TSG understanding and application through RAG, (2) offering tailored recommendations for specific industry needs, and (3) providing a comprehensive benchmarking platform for users to assess and select suitable TSG methods and metrics. These scenarios demonstrate how TSGAssist effectively mitigates the challenges faced by industry professionals in navigating the complex landscape of TSG, fostering greater application of TSG in various industrial domains.

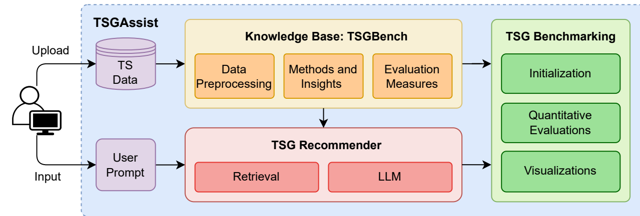


Figure 1: The system architecture of TSGAssist.

2 PRELIMINARIES

2.1 Time Series Generation

The primary goal of Time Series Generation (TSG) is to generate synthetic time series data that is indistinguishable from the real data, preserving the underlying patterns and correlations. TSG plays a crucial role in applications like anomaly detection [3] and privacy preservation [7]. Contemporary approaches in TSG predominantly employ three foundational generative models: Generative Adversarial Networks (GANs), Variational AutoEncoders (VAEs), and Flow-based models. GAN-based models, like TimeGAN [13] and RTSGAN [11], capture temporal dependencies through adversarial training. VAE-based methods, such as TimeVAE [5] and TimeVQVAE [8], generate new time series using latent representations, ensuring data fidelity. Lastly, Flow-based models [1] use transformations to effectively model complex data distributions.

In terms of evaluation, TSGBench emerges as a key benchmarking framework for TSG methods, comprising three core parts:

- **Data Preprocessing:** TSGBench provides a standardized pipeline for preprocessing real-world time series data, including normalization and segmentation, to ensure uniformity and comparability across different TSG methods.
- **Methods and Insights:** TSGBench offers insights and guidance for selecting TSG methods based on data characteristics and needs. It highlights the significance of factors like data dimensionality, sequence length, and application domain.
- **Evaluation Measures:** TSGBench introduces a suite of evaluation metrics, including model-based, feature-based, and distance-based measures. These metrics provide a comprehensive overview of each method's performance.

2.2 LLMs and RAG

Large Language Models (LLMs). LLMs have recently become a transformative force, excelling in understanding and generating natural language due to their training on extensive text corpora [4]. These models handle intricate linguistic structures and semantic nuances effectively. LLMs operate through natural language prompts and can engage in multi-round dialogues, considering previous interactions to provide more contextual and accurate responses [4]. This capability is particularly useful in applications like TSGAssist, transforming it into an automated conversational agent rather than a rule-based system. In practice, this allows users to interactively obtain TSG method recommendations for specific datasets, streamlining the process beyond traditional manual searches.

Retrieval-Augmented Generation (RAG). RAG [9] enhances LLMs by enabling them to utilize external knowledge bases during generation, fetching relevant information for user queries. In the RAG process, upon receiving a prompt, the system searches the related knowledge base, extracts relevant data, and integrates it into the LLM's response, improving accuracy and reducing content hallucinations [9]. In the context of TSGAssist, when users inquire about specific TSG methods, RAG-equipped LLMs quickly access detailed information, enriching and validating interactions. This capability makes RAG-enhanced LLMs crucial for TSG applications, providing depth of knowledge and decision-making support.

3 TSGASSIST

We introduce TSGAssist, a standalone web application developed with Python 3.7 and the Dash framework [6], providing a user-friendly GUI. Figure 1 depicts its architecture, and Figure 4 displays its core functions through screenshots. It includes basic functions like a control panel for data upload, TSG method and measure selection (Figure 4(a.1)), along with a Time Series Overview canvas for data selection and raw MTS visualization (Figure 4(a.2)). There are two principal modules in TSGAssist: (1) **TSG Recommender** for personalized method suggestions, and (2) **TSG Benchmarking** for hands-on evaluation and comparison of TSG methods.

3.1 TSG Recommender

TSG Recommender offers personalized suggestions via a chatbot-style interface, allowing users to interact with TSGAssist. As shown in Figure 2, it operates through a series of interconnected components, employing RAG and LLMs to harness insights from TSGBench for user-specific recommendations. Specifically,

- **User Prompt**: Users begin by interacting with the system, entering TSG-related queries, and uploading datasets.
- **Data Preprocessing**: Upon receiving the user's input, this component preprocesses the time series data to better grasp the context and specifics of each request.
- **Knowledge Base**: Concurrently, the extensive TSGBench knowledge base is accessed, which includes detailed information on TSG methods, metrics, and datasets essential for up-to-date and relevant recommendations.
- **Retrieval**: The core of TSG Recommender is Retrieval, powered by RAG and LLMs and optimized to effectively search the knowledge base to answer user queries. Using GPT-4[1] [10], it retrieves pertinent, timely information from TSGBench.
- **LLM**: After retrieval, this component processes the user's prompt and fetched data to generate relevant responses. The LLM's ability to understand and produce natural language is crucial for a seamless and engaging user experience.

[1] https://platform.openai.com/docs/assistants/overview

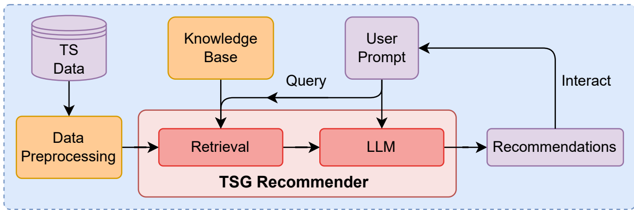
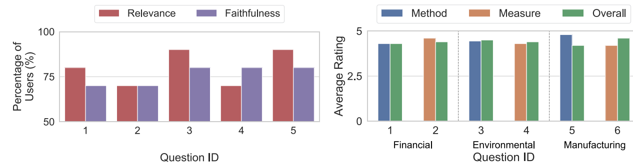


Figure 2: The internal functions of TSG Recommender.



(a) RAG Efficacy. (b) Domain-Specific Applications.

Figure 3: Demonstration Scenarios 1 and 2.

- **Recommendations**: Finally, the system delivers its recommendations or information to the user.

TSG Recommender supports continuous dialogue, letting users iteratively adjust their queries in the same conversation. This capability fosters in-depth exploration and clarification, thereby improving the decision-making process with more precise recommendations.

3.2 TSG Benchmarking

Guided by TSG Recommender, the TSG Benchmarking module allows users to assess the suggested TSG methods and measures suited to their specific tasks. This module enables users to apply these methods to their datasets and investigate results using sample datasets, enhancing their understanding. It has three components:

- **Initialization**: Users first choose recommended TSG methods and evaluation metrics from the control panel and then proceed to the benchmarking platform. This ensures evaluations are targeted and pertinent.
- **Quantitative Evaluations**: It includes a broad spectrum of metrics from TSGBench, covering model-based, feature-based, and distance-based categories, and tracks each method's running time for efficiency insights. This comprehensive evaluation allows users to assess TSG methods from multiple perspectives thoroughly. The chosen metrics are clearly displayed on the main interface, with options for detailed analysis.
- **Visualizations**: This component provides visualization tools like PCA, t-SNE, and Distribution Plot on the secondary tier for comparing original and generated time series.

This module's innovative feature is its dynamic update system. Whenever a new TSG method is selected, it automatically updates the evaluation results and visualizations, ensuring the module remains current and in sync with the latest developments.

4 DEMONSTRATION SCENARIOS

This demonstration unfolds across three meticulously crafted scenarios, each spotlighting a distinct facet of TSGAssist's capabilities. The primary goals are to: (1) showcase the efficacy of integrating RAG in the TSG Recommender, (2) offer domain-specific recommendations for TSG methods and evaluation measures, and (3) facilitate thorough evaluation of various TSG methods.

4.1 Scenario 1: RAG Efficacy

The first scenario establishes the foundational effectiveness of TSG Recommender's RAG component.

- **Objective:** The goal is to show the improvement in the relevance and faithfulness of responses from an LLM when augmented with RAG, as opposed to those by the LLM alone.
- **Setup:** We engaged 10 participants, who were given a series of prompts to input into two systems: a standalone LLM and a RAG-enhanced LLM in TSGAssist. The responses from both systems were displayed side by side anonymously, with participants unaware of each system's contribution.
- **Questions:** (1) *Relevance:* Participants assessed which responses are more relevant to their initial prompt, gauging the system's ability to effectively understand and address the query. (2) *Faithfulness:* Participants were asked to evaluate which of the two responses appears to be more consistent with existing knowledge in TSGBench, thereby assessing the faithfulness of the information provided.

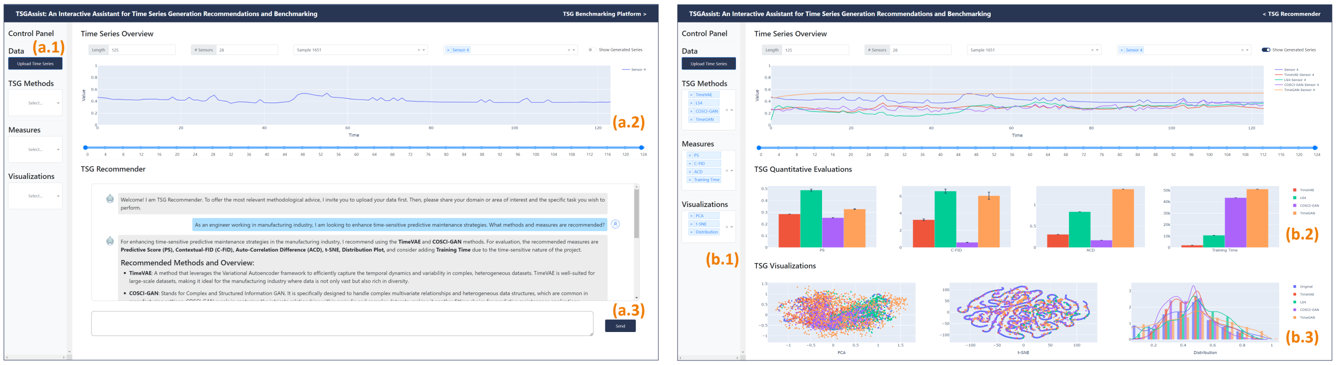
Results from Figure 3(a) indicate that over 70% of users found the RAG-enhanced LLM in TSG Recommender to be significantly more effective in terms of relevance and faithfulness than the standalone LLM. The RAG-enhanced system's responses were more aligned with the TSGBench knowledge base, showing a higher factual accuracy. Moreover, responses from TSGAssist were rated as more pertinent to user prompts, suggesting a better understanding of user queries. These results affirm the benefits of integrating RAG with LLMs, enhancing the accuracy and relevance of generated text, essential for the TSG Recommender.

4.2 Scenario 2: Domain-Specific Applications

Building on the success of the RAG component in Scenario 1, Scenario 2 emphasizes its performance in domain-specific contexts, underscoring its adaptability and precision in providing tailored recommendations for TSG methods and evaluation measures.

- **Objective:** The aim is to showcase TSGAssist's capability to generate recommendations that are not only methodologically sound but also contextually relevant to specific domains.
- **Setup:** Participants from finance, environmental science, and manufacturing are provided with responses generated by TSG Recommender tailored to each domain.
- **Questions:** For each question, participants were instructed to rate the recommendations using a 1 to 5 scale based on either (1) *Method Comprehensiveness* or (2) *Measure Appropriateness*, along with (3) *Overall Domain Helpfulness*. In particular, (1) Method Comprehensiveness: Participants evaluated the applicability of recommended TSG methods in their field. (2) Measure Appropriateness: Participants rated the relevance of TSG evaluation measures for their domain. (3) Overall Domain Helpfulness: Participants scored the overall effectiveness of TSGAssist in meeting their specific domain needs.

As evidenced by the feedback shown in Figure 3(b), the average rating for all questions across various domains consistently



(a) TSG Recommender.

(b) TSG Benchmarking Platform.

Figure 4: Screenshots of TSGAssist.

exceeded 4.2. The TSG methods and metrics recommended by TSGAssist were highly valued for their relevance and practicality in specific domains. Overall, the responses affirmed TSGAssist's effectiveness in addressing domain-specific queries.

4.3 Scenario 3: TSG Benchmarking Platform

The third scenario showcases TSGAssist as a holistic TSG benchmarking platform for both academic and industry professionals. For example, a user from the manufacturing sector uploads sensor-based time series into TSGAssist (Figure 4(a.1)), seeking optimal TSG methods for their data traits and predictive maintenance needs.

Recommendations: TSG Recommender (Figure 4(a.3)) advises starting with VAE-based methods like TimeVAE [5] and TimeVQVAE [8] for TSG. It also suggests COSCI-GAN [12] for its proficiency in handling complex multivariate relationships, crucial in machinery interactions. For evaluation, TSGAssist recommends using Predictive Score (PS) to compare models trained on synthetic and original data for maintenance predictions. It also endorses C-FID and ACD to verify the preservation of temporal correlations in synthetic data, reflecting real machinery operation patterns.

Benchmarking: Users begin on the benchmarking platform by selecting recommended methods and metrics from the control panel (Figure 4(b.1)). Comparative results (Figure 4(b.2)) highlight TimeVAE and COSCI-GAN as top choices, noting TimeVAE's significantly faster training compared to GAN-based alternatives. Additional methods like TimeGAN can be included for a more thorough analysis. Users can compare synthetic and original data by PCA, t-SNE, or Distribution Plot (Figure 4(b.3)), further confirming the suitability and effectiveness of the chosen methods.

User Experience and Impact: Manufacturing managers and professionals can seamlessly integrate recommended TSG methods and evaluation metrics into their workflows using TSGAssist without delving into extensive TSG literature. Its accessibility and practicality significantly simplify the adoption and integration of advanced TSG solutions into real-world applications, improving predictive maintenance strategies and operational efficiency.

5 CONCLUSIONS

In this demonstration, we develop TSGAssist, an innovative system that bridges the gap between advanced TSG research and practical industry application. Utilizing the comprehensive capabilities of TSGBench, it offers industry professionals an accessible platform for customized TSG recommendations and interactive, user-friendly benchmarking across diverse evaluation measures.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative and the Ministry of Education, Singapore, under its MOE AcRF TIER 3 Grant (MOE-MOET32022-0001). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore and the Ministry of Education, Singapore.

REFERENCES

[1] Ahmed M. Alaa, Alex James Chan, and Mihaela van der Schaar. 2021. Generative Time-series Modeling with Fourier Flows. In *ICLR*.
[2] Yihao Ang, Qiang Huang, Yifan Bao, Anthony KH Tung, and Zhiyong Huang. 2023. TSGBench: Time Series Generation Benchmark. *Proc. VLDB Endow.* 17, 3 (2023), 305–318.
[3] Yihao Ang, Qiang Huang, Anthony KH Tung, and Zhiyong Huang. 2023. A Stitch in Time Saves Nine: Enabling Early Anomaly Detection with Correlation Analysis. In *ICDE*. 1832–1845.
[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*. 1877–1901.
[5] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. 2021. TimeVAE: A Variational Auto-Encoder for Multivariate Time Series Generation. arXiv:2111.08095
[6] Plotly Technologies Inc. 2021. Dash. https://plotly.com/dash/.
[7] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*.
[8] Daesoo Lee, Sara Malacarne, and Erlend Aune. 2023. Vector Quantized Time Series Generation with a Bidirectional Prior Model. In *AISTATS*. 7665–7693.
[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*. 9459–9474.
[10] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774
[11] Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. 2021. Towards generating real-world time series data. In *ICDM*. 469–478.
[12] Ali Seyfi, Jean-François Rajotte, and Raymond T. Ng. 2022. Generating multivariate time series with COmmon Source CoordInated GAN (COSCI-GAN). In *NeurIPS*. 32777–32788.
[13] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. 2019. Time-series Generative Adversarial Networks. In *NeurIPS*. 5509–5519.